PadhAI: 6 Jars of Sigmoid Neuron

One Fourth Labs

Writing the code

How do we implement this in Python

1. Here is the Python code for Gradient descent

```python
X = [0.5, 2.5]
Y = [0.2, 0.9]

def f(w, b, x):
    #Sigmoid with parameters w and b
    return 1.0 / (1.0 + np.exp(-(w*x + b)))

def error(w, b):
    err = 0.0
    for x,y in zip(X, Y):
        fx = f(w, b, x)
        err += 0.5 * (fx - y) ** 2
    return err

def grad_b(w, b, x, y):
    fx = f(w, b, x)
    return (fx - y) * fx * (1 - fx)

def grad_w(w, b, x, y):
    fx = f(w, b, x)
    return (fx - y) * fx * (1 - fx) * x

def do_gradient_descent():
    w, b, eta = 0, -8, 1.0
    max_epochs = 1000
    for i in range(max_epochs):
        dw, db = 0, 0
        for x, y in zip(X, Y):
            dw += grad_w(w, b, x, y)
            db += grad_b(w, b, x, y)
        w = w - (eta * dw)
        b = b - (eta * b)
```

2. This is how the algorithm works